Amendments to the Claims

1. (Original) A system for intercepting one or more application program interface (API) calls in a virtual memory environment comprising: an activation module and an interception module:

said activation module being adapted to load said interception module to occupy a location in a shared region of virtual memory as long as interception of said API calls is required;

said activation module being adapted to redirect said one or more API calls by creating an alias to any page containing an entry point for an API call to be intercepted and to write the address of said interception module to said alias;

said activation module being adapted to provide to any instances of said interception module the original entry points for said one or more API calls; and

said interception module being adapted to selectively provide modified functionality for said intercepted API calls.

2. (Original) A system as claimed in claim 1 wherein said activation module is adapted to load said interception module at system initialization time.

3. (Original) A system as claimed in claim 1 wherein said activation module is adapted to read a configuration file containing data defining the API calls to be intercepted and the manner in which said API calls are to be modified, said activation module being adapted to write said configuration data to an area of shared memory and to provide to all instances of said interception module the location of said shared memory.

4. (Original) A system as claimed in claim 3 wherein an instance of said interception module is responsive to receiving a redirected API call to load itself within the process

making said API call so that a segment of global data is made available to each instance of said interception module.

5. (Previously presented) A system for intercepting one or more application program interface (API) calls in a virtual memory environment comprising: an activation module and an interception module:

said activation module being adapted to load said interception module to occupy a location in a shared region of virtual memory as long as interception of said API calls is required;

said activation module being adapted to redirect said one or more API calls by creating an alias to any page containing an entry point for an API call to be intercepted and to write the address of said interception module to said alias;

said activation module being adapted to provide to any instances of said interception module the original entry points for said one or more API calls; and

said interception module being adapted to selectively provide modified functionality for said intercepted API calls;

said activation module being adapted to read a configuration file containing data defining the API calls to be intercepted and the manner in which said API calls are to be modified;

said activation module being adapted to write said configuration data to an area of shared memory and to provide to all instances of said interception module the location of said shared memory;

an instance of said interception module being responsive to receiving a redirected API call to load itself within the process making said API call so that a segment of global data is made available to each instance of said interception module; and

said activation module being adapted to write the location of said shared memory to said global data segment and wherein any instance of said interception module is adapted to use said location to read said configuration data from shared memory.

6. (Currently amended) A system ~~as claimed in claim 1~~ for intercepting one or more application program interface (API) calls in a virtual memory environment comprising: an activation module and an interception module:

said activation module being adapted to load said interception module to occupy a location in a shared region of virtual memory as long as interception of said API calls is required;

said activation module being adapted to redirect said one or more API calls by creating an alias to any page containing an entry point for an API call to be intercepted and to write the address of said interception module to said alias;

said activation module being adapted to provide to any instances of said interception module the original entry points for said one or more API calls;

said interception module being adapted to selectively provide modified functionality for said intercepted API calls;

~~wherein~~ said interception module ~~is~~ being adapted to export one or more global variables located in its code segment to said activation module, each global variable corresponding to an API call to be intercepted~~,~~; and

said activation module being adapted to create an alias to any page containing one of said one or more global variables to write the original entry point of said one or more API calls to said respective aliases.

7. (Previously presented) A system for intercepting one or more application program interface (API) calls in a virtual memory environment comprising: an activation module and an interception module:

said activation module being adapted to load said interception module to occupy a location in a shared region of virtual memory as long as interception of said API calls is required;

said activation module being adapted to redirect said one or more API calls by creating an alias to any page containing an entry point for an API call to be intercepted and to write the address of said interception module to said alias;

said activation module being adapted to provide to any instances of said interception module the original entry points for said one or more API calls;

said interception module being adapted to selectively provide modified functionality for said intercepted API calls; and

said activation module being adapted to operate in at least two modes, an instance running in a first mode being adapted to instantiate an instance to run in a second mode wherein said instance running in said second mode runs as a daemon so maintaining said interception module in virtual memory.

8. (Original) A system as claimed in claim 7 wherein each instance of said activation module is adapted to intercommunicate through a shared read/write communications segment.

9. (Original) A system as claimed in claim 5 wherein said interception module is adapted to utilize a spin lock variable located in said global data segment to serialize access to said configuration data.

10. (Original) A system as claimed in claim 5 wherein said interception module is adapted to use an exception handler to access said configuration data.

11. (Original) A system as claimed in claim 4 wherein said interception module is adapted to check that it is not being called recursively prior to loading itself.

12. (Original) A system as claimed in claim 1 wherein one of said one or more API calls to be intercepted is a call for allocating memory.

13. (Original) A system as claimed in claim 1 wherein said activation module and said interception module are adapted to operate on OS/2 Warp Version 3 SMP and Warp Version 4.5 operating systems.

14. (Original) A computer program product comprising computer program code stored on a computer readable storage medium for intercepting API calls when executed on a virtual memory computer system, the program code comprising the system of claim 1.

15. (Previously presented) A computer program product comprising computer program code stored on a computer readable storage medium for intercepting API calls when executed on a virtual memory computer system, the program code comprising the system of claim 5.

16. (Previously presented) A computer program product comprising computer program code stored on a computer readable storage medium for intercepting API calls when executed on a virtual memory computer system, the program code comprising the system of claim 7.

17. (New) A method for intercepting one or more application program interface (API) calls in a virtual memory environment, comprising the steps of:

loading an interception module selectively providing modified functionality for said intercepted API calls to occupy a location in a shared region of virtual memory as long as interception of said API calls is required;

redirecting said one or more API calls by creating an alias to any page containing an entry point for an API call to be intercepted and writing the address of said interception module to said alias; and

providing to any instances of said interception module the original entry points for said one or more API calls.